



# Исполнители

# РОБОТЛАНДИИ



**Исполнитель**- это модель реального устройства или действия (процесса).

*Исполнитель* является помощником человека в решении важных практических задач.

Примеры исполнителей Роботландии:

Перевозчик, Конюх, Переливашка, Курсор, Квадратик, Автомат...



# Система команд исполнителя (СКИ).

1. Обозначение(название) команды;
2. Способ передачи её исполнителю;
3. Выполнение команды(как она выполняется).



## Два вида ошибок:

***НЕ ПОНИМАЮ-*** синтаксические ошибки,

Это ошибки формальной записи некоторых команд-команда не принадлежит СКИ.

***НЕ МОГУ-*** семантические ошибки.

Это ошибки, заставляющие Исполнителя выполнять действия, которые выходят за пределы его возможностей.

Они специфичны не только для каждого Исполнителя, но и для каждой команды.

**Ошибки совершает не робот, а человек, который им управляет!**



# Схема знакомства с исполнителем





# Исполнитель КВАДРАТИК

## Среда:

Поле, в котором исполнитель может рисовать различные фигуры

## Названия команд:

Треугольник, Квадрат, Круг, Закраска, Очистка.



# Основные исполнители Роботландии.

Машинист,

Буквоед,

Плюсик.



# ИСПОЛНИТЕЛЬ «Машинист»

## СКИ

- ВПЕРЁД
- НАЗАД
- ОТЦЕПИ
- ПРИЦЕПИ
- СТРЕЛКА



# ЗАДАЧА:



**Переставить состав на нижний путь и вернуть локомотив на место.**

**Дано:**

ВЦП



М

**Надо:**

В ЦП



М

**Решение:**

# Алгоритм (схема).



Дано:

Надо:

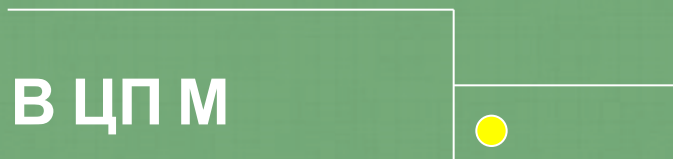
1. Переставить состав на нижний путь:

ВЦП



2. Вернуть локомотив на место:

В ЦП М



В ЦП





# Алгоритм (СКИ):

- |    |           |    |          |
|----|-----------|----|----------|
| 1) | ■ Вперед  | 2) | ■ Отцепи |
|    | ■ Вперед  |    | ■ Отцепи |
|    | ■ Прицепи |    | ■ Отцепи |
|    | ■ Прицепи |    | ■ Назад  |
|    | ■ Прицепи |    |          |
|    | ■ Назад   |    |          |
|    | ■ Вперед  |    |          |
|    | ■ Стрелка |    |          |
|    | ■ Вперед  |    |          |

# ЗАДАЧА



Дано:

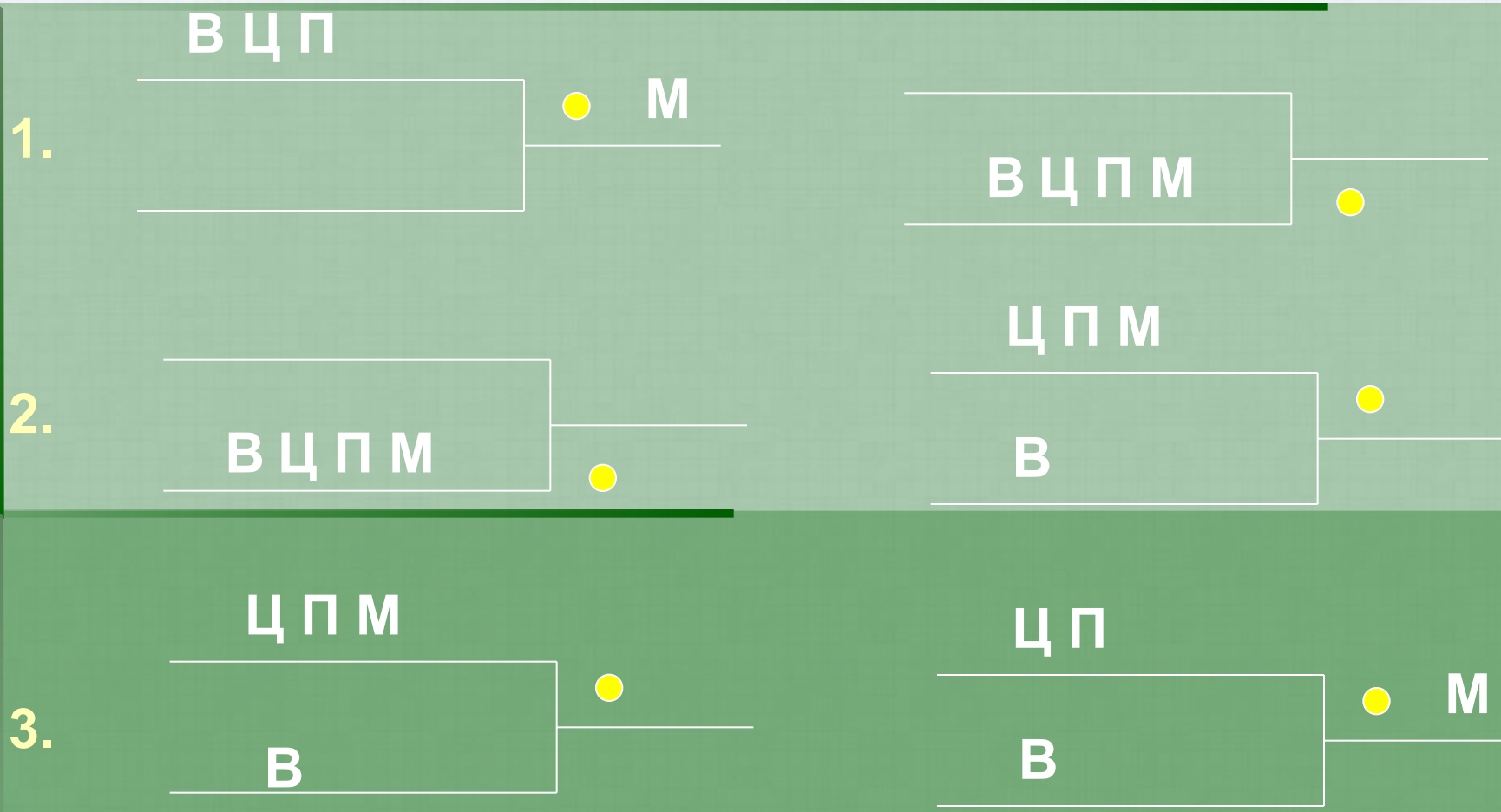
Надо:



Составить алгоритм перестановки на нижнюю ветку первого вагона (В) состава (самого дальнего от локомотива) и возвращения локомотива в исходную позицию.



# Решение (алгоритм)





# Алгоритм (СКИ)

1) Переставить состав на нижний путь:

Вперед  
Вперед  
Прицепи  
Прицепи  
Прицепи  
Назад  
Вперед  
Стрелка  
вперед



2) Переставить пару «цистерна-платформа» на верхний путь:

Отцепи  
Назад  
Вперед  
Стрелка  
Вперед

3) Вернуть локомотив на место:

Отцепи  
Отцепи  
Назад



2) Переставить пару  
«цистерна-платформа» на  
верхний путь:

Отцепи  
Назад  
Вперед  
Стрелка  
Вперед

3) Вернуть локомотив на  
место:

Отцепи  
Отцепи  
Назад





# ЗАДАЧА:

(Д/З)

**Переставить крытый вагон на первое место в составе.**

Дано:

В Ц П



Надо:





# Исполнитель АВТОМАТ.

**СКИ:**

**ПРИБАВЬ** - *Прибавляет единицу;*

**УМНОЖЬ** - *Умножает на два (удваивает)*

**Оптимальный алгоритм**- алгоритм решения задачи с наименьшим числом команд.

# Задача. Составить оптимальный алгоритм получения числа **69**.

Решение.



<i>Число</i>	<i>Какой командой получено</i>	<i>Из какого числа</i>	<i>Номер команды</i>
<b>69</b>	<b>Прибавь</b>	<b>68</b>	<b>9</b>
<b>68</b>	<b>Умножь</b>	<b>34</b>	<b>8</b>
<b>34</b>	<b>Умножь</b>	<b>17</b>	<b>7</b>

<b>17</b>	<b>Прибавь</b>	<b>16</b>	<b>6</b>
<b>16</b>	<b>Умножь</b>	<b>8</b>	<b>5</b>
<b>8</b>	<b>Умножь</b>	<b>4</b>	<b>4</b>
<b>4</b>	<b>Умножь</b>	<b>2</b>	<b>3</b>
<b>2</b>	<b>Умножь</b>	<b>1</b>	<b>2</b>
<b>1</b>	<b>Прибавь</b>	<b>0</b>	<b>1</b>



# Проверочная работа.

Составить оптимальный алгоритм получения числа

I вариант

37

II вариант

47



# Исполнитель с памятью ПЛЮСИК.

## СКИ:

Запомни  $x$  ( $x$ - параметр)

Сложи

Вычти

Умножь

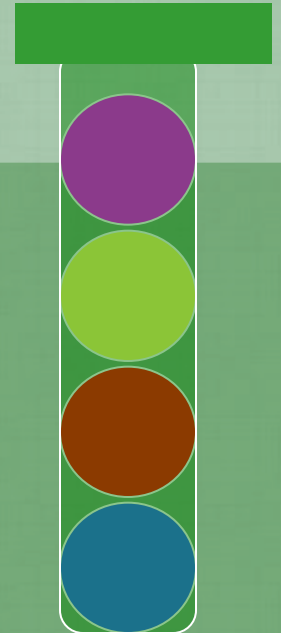
Дели

Очисти



# Способ организации памяти-стек.

*Стеком* называется *хранилище информации*, в котором доступным элементом является последний поступивший в хранилище.





**Правило вычитания:**

**Уменьшаемое должно попасть в стек раньше вычитаемого!!!**

*Правило деления:*

**Делимое** должно попасть в стек раньше **делителя!!!**